IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | | | |
|---|---|---|---|---|
| Appellants: | Daniel N. CRIPE et al. | §<br>§ | Confirmation No.: | 8712 |
| Serial No.: | 10/717,730 | §<br>§ | Group Art Unit: | 2416 |
| Filed: | 11/20/2003 | §<br>§ | Examiner: | P. Sinkantarakorn |
| For: | Method And System Of Teamed Network Adapters With Offloaded Connections | §<br>§<br>§<br>§ | Docket No.: | 200313587-1 |

# APPEAL BRIEF

**Mail Stop Appeal Brief – Patents** Date: September 29, 2009
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal is electronically filed concurrently herewith.

## TABLE OF CONTENTS

## I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company, L.P. (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas. HPDC is a wholly-owned affiliate of Hewlett-Packard Company (HPC). The Assignment from the inventors to HPDC was recorded on November 20, 2003 at Reel/Frame 014724/0976.

## II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

## III. STATUS OF THE CLAIMS

| | |
|---|---|
| Originally filed claims: | 1-21. |
| Claim cancellations: | None. |
| Added claims: | None. |
| Presently pending claims: | 1-21. |
| Presently appealed claims: | 1-21. |

## IV. STATUS OF AMENDMENTS

Appellants have not filed any amendments after the Final Office Action dated July 27, 2009.

## V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Various embodiments of the invention are described below. The scope of disclosure is not limited by the descriptions of the embodiments that follow. Citations to the specification have been provided to demonstrate where support may be found in the specification for various parts of the invention. Additional support may be found elsewhere in the application.

Appellants' contribution is directed to a technique for offloading and reloading network connections among multiple Network Interface Cards (NICs) 118, 120, 122 that have been teamed together. Specifically, a server 102 may communicate with one or more clients 104, 106 via a network switch 124. P. 3, ll. 21-23; Figure 1. The server 102 communicates with the network switch 124 using a plurality of NICs 118, 120, 122. P. 4, ll. 1-2; Figure 1. Processing of a TCP/IP stack 114 (used to establish and maintain network connections) in the server 102 is performed by a CPU 108, but may be "offloaded" (or "moved") to circuit logic in one of the NICs 118, 120, 122. P. 5, l. 31 – p. 6, l. 5; Figures 1-2. When NICs are operating in a team, they share common addresses. P. 6, ll. 22-24; Figure 1. Thus, packets received by the team may be received by any one of the NICs in the team. P. 3, ll. 30-31; Figure 1. When a NIC in the team becomes inoperative (*e.g.*, due to failure), another NIC in the team may receive the packet. P. 8, ll. 4-8; Figure 1. The technique mentioned above comprises detecting the receipt of a packet on a different-than-usual NIC and, as a result, offloading a network connection from the defective NIC to the NIC on which the packet was received. P. 9, ll. 8-16; Figures 1 and 4. Clearly, therefore, this offloading from the defective NIC to the NIC on which the packet was received **is precipitated by the receipt of the packet on that non-defective NIC**.

Claim 1 is directed to a computer system 100 that comprises a central processing unit (CPU) 108 and first and second network adapters 118, 120, 122 teamed together and configured to receive offloaded connections. P. 5, l. 31 – p. 6, l. 5; p. 6, ll. 22-24; Figure 1. A program 116 executing on the CPU 108 reloads an offloaded connection established by the first network adapter 118, 120, 122 onto the second network adapter 118, 120, 122 as a result of one of a plurality of packets associated with the offloaded connection being received on the second network adapter 118, 120, 122. P. 9, ll. 8-16; Figures 1 and 4.

Claim 8 is directed to a method that comprises examining a packet received from an external device 124 and determining whether a connection associated with the packet is currently offloaded. Col. 8, l. 29 – col. 9, l. 3; Figures 1 and 4. The method also comprises reloading the connection in response to the packet associated with the connection being offloaded and received by a network interface 118, 120, 122 not currently processing the offloaded connection. P. 9, ll. 8-16; Figures 1 and 4.

Claim 12 is directed to a computer readable media 110 storing instructions 116 executable by a computer system 100, and when executed the instructions implement a method that comprises examining a packet received from an external device 124 and determining whether a connection 118, 120, 122 associated with the packet is currently offloaded. Col. 8, l. 29 – col. 9, l. 3; Figures 1 and 4. The method also comprises reloading the connection as a result of the packet associated with the connection being offloaded and received by a network interface 118, 120, 122 not currently processing the offloaded connection. P. 9, ll. 8-16; Figures 1 and 4.

Claim 16 is directed to a computer system 100 comprising means (*e.g.*, CPU 108; Figure 1) for reading and executing programs. Figure 1. The system 100 also comprises first and second means (*e.g.*, NICs 118, 120, 122; Figure 1) for sending and receiving data connections over a network, where the first and second means are grouped together and are capable of processing offloaded data connections. P. 3, ll. 21-23; p. 5, l. 31 – p. 6, l. 5; Figure 1. A program executed by the means for reading and executing programs reloads an offloaded connection established by the first means for sending and receiving data onto the second means for sending and receiving data in response to one of a plurality of packets associated with the offloaded connection being received on the second means for sending and receiving data. P. 9, ll. 8-16; Figures 1 and 4.

## VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the Examiner erred in rejecting claims 1-4 and 6-21 under 35 U.S.C. § 103(a) as obvious in view of Congdon (U.S. Pat. No. 6,151,297) and Burns (U.S. Pat. No. 6,938,092).

Whether the Examiner erred in rejecting claim 5 under 35 U.S.C. § 103(a) as obvious in view of Congdon, Burns and Mahalingham (U.S. Pat. No. 6,314,525).

## VII. ARGUMENT

### A. Summary of Burns

The Examiner cites col. 6, l. 36 – col. 7, l. 67 of Burns as relevant. Col. 6, ll. 36-67 relate to Burns, Figure 2. Figure 2 shows a Network Interface Device (NID) 103 that sends and receives data to and from a hub/switch 105 via ports 1 and 2. Port 1 transfers data from the hub/switch 105 to NID 103; port 2 transfers data from the NID 103 to the hub/switch 105. Port 1 is associated with MAC address "A," while port 2 is associated with MAC address "C." Both ports 1 and 2 are said to be associated with "connection #1." Burns teaches that when a failure occurs on Port 1, the NID 103 detects the failure and notifies a port aggregation driver 112. In turn, the driver 112 updates its data structures accordingly. Ports 1 and 2 are part of the same team. Port 1 may be designated as the primary port. When port 1 fails, the driver 112 chooses port 2 to be the new, primary team member by adjusting a pointer as shown in Figure 3.

**Despite these teachings in col. 6, ll. 36-67, Burns does not teach that port 2 is selected as a result of receiving a packet associated with the connection previously coupled to port 1.**

The relevant portions of col. 7, ll. 1-56 generally discuss the precise manner in which MAC switching is performed. They also discuss the NDIS function, which is mentioned in Section VII(B)(3) of this Appeal Brief. Col. 7, ll. 57-67 further describes what occurs if port 1 fails. In particular, if port 2 is to continue transmitting for connection #1, and if port 2's MAC address is to be changed from MAC C to MAC A, the driver 112 updates the TCB 119 for connection #1 by changing the MAC source address. **Again, however, Burns does not teach or even suggest that port 2's MAC address is changed or updated as a result of receiving a packet associated with the connection previously coupled to port 1.**

### B. The Examiner erred in rejecting claims 1-4 and 6-21 at least because the combination of Congdon and Burns fails to disclose all claim limitations

The Examiner rejected claims 1-4 and 6-21 as allegedly obvious in view of Congdon and Burns. Appellants respectfully traverse this rejection. Independent claim 1 is representative of this grouping of claims. The grouping should not be construed to mean the patentability of any of the claims may be determined in later actions (*e.g.*, actions before a court) based on the groupings. Rather, the presumption of 35 USC § 282 shall apply to each of these claims individually.

#### 1. Summary of recent prosecution history

Claim 1 requires "wherein a program executing on the CPU reloads an offloaded connection established by the first network adapter onto the second network adapter **as a result of one of a plurality of packets associated with the offloaded connection being received on the second network adapter**" (emphasis added). In the Office Action dated May 16, 2008, the Examiner conceded that the emphasized portion of the limitation cited above is not found in Congdon. As a result, the Examiner turned to Burns and asserted that col. 6, ll. 36-67 and col. 7, ll. 57-67 of Burns satisfies Congdon's deficiencies.

Appellants filed an Appeal Brief on October 6, 2008, conclusively demonstrating why these portions of Burns fail to satisfy Congdon's deficiencies. In light of this Appeal Brief, the Examiner reopened prosecution. In an Office Action dated January 15, 2009, the Examiner cited col. 7, ll. 1-29 of Burns in addition to the previously-cited portions of Burns as satisfying Congdon's deficiencies. Appellants filed a response on April 14, 2009 clearly explaining why the newly-cited portions of Burns still fail to teach the quoted limitation. On July 27, 2009, the Examiner issued yet another Final Office Action sustaining his prior rejection and providing additional arguments. This Appeal Brief is being filed in response to that Final Office Action.

#### 2. Plea to Board

As a preliminary note, Appellants respectfully ask the Board to note that this is Appellants' **third** appeal brief. Examiner Sinkantarakorn persists in rejecting the claims using the same references when Appellants have repeatedly established why the references fail to anticipate the claims or render the claims obvious. The Examiner's numerous and improper rejections are delaying a timely conclusion to the prosecution process and are burdening Appellants with additional expense. Appellants' pleas to the Examiner have been unfruitful. Thus, Appellants very respectfully ask the Board to carefully consider the pending claims and reverse the Examiner's rejections so that Appellants may finally conclude prosecution on this matter.

### 3. Appellants' basic argument in Response to Office Action Dated April 14, 2009

The relevant portions of Burns (col. 6, l. 36 – col. 7, l. 67) cited by the Examiner discuss port redundancy in reference to Fig. 2 (reproduced below for the Examiner's convenience).

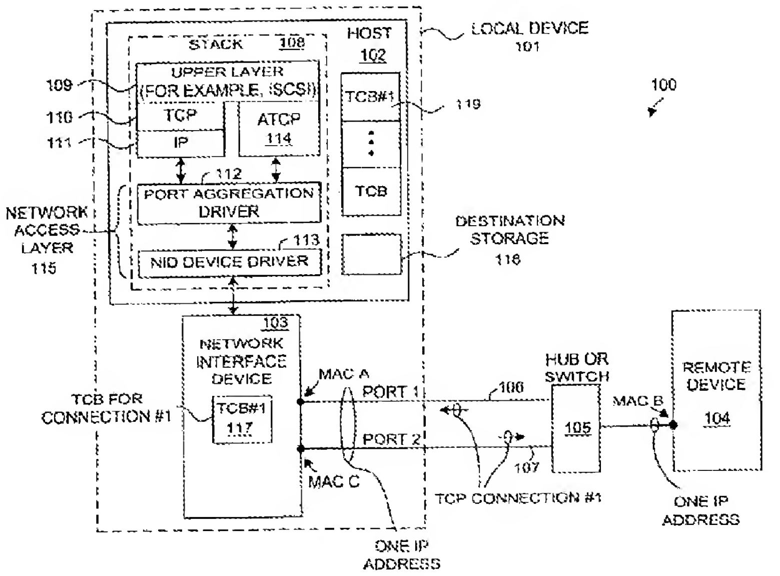

FIG. 2

In particular, Burns teaches that the network interface device (NID) 103 sends and receives data from the switch 105 via ports 1 and 2. *See* col. 4, ll. 13-33 and Fig. 2. Port 1 is assigned MAC address "A" while port 2 is assigned MAC address "C." Col. 7, ll. 10-13; Fig. 2. Burns notes that one of the ports may fail.

Col. 6, l. 36. For example, port 1 may fail and thus may be unable to transfer data between NID 103 and switch 105. Burns teaches that the NID 103 detects such failure (col. 6, ll. 40-41) and sends a signal to the port aggregation driver (PAD) 112 indicating the failure. Col. 6, ll. 42-52.

In response to receiving this failure signal from the NID 103, the PAD 112 "chooses another of the ports of the team (port 2 in this example) to be the new 'primary' team member." Col. 6, ll. 61-67. The PAD 112 further swaps the MAC addresses of the ports, so that the current MAC address assignments (port 1 with MAC address "A" and port 2 with MAC address "C") are reversed (port 1 with MAC address "C" and port 2 with MAC address "A").

Significantly, Burns teaches that the PAD 112 swaps MAC addresses by writing MAC "A" into the MAC address field for port 2 and by writing MAC "C" into the MAC address field for port 1. Col. 7, ll. 6-9. Burns teaches that the PAD 112 causes the NID 103 to change MAC addresses for the ports using the NDIS request function to which the Examiner cites in the instant Office Action. Col. 7, ll. 10-29. Similarly, col. 7, ll. 57-67 echo this theme of changing MAC addresses, except in this case, the PAD 112 updates the TCB 119.

Despite these teachings regarding MAC address switches, Burns certainly does not teach or even begin to suggest that a MAC address is switched "as a result of one of a plurality of packets associated with the offloaded connection being received on the second network adapter," as required by claim 1. In order for Burns to teach such a limitation, Burns would have to teach that the MAC switching described in cols. 6-7 occurs as a result of the non-failed port (port 2) receiving a data packet associated with the failed port (port 1). Instead, Burns teaches that the MAC addresses are switched as a result of the NID 103's detection of port failure and the subsequent failure signal sent to the PAD 112. There is no teaching in Burns that the receipt of a packet – any packet – precipitates MAC address swapping.

Having established that Burns fails to teach the limitation in question, Appellants now turn to the Examiner's specific argument. The Examiner appears to believe that Burns teaches the limitation in question and offers the following line of reasoning for support: As a result of port 1's failure, port 2 becomes a primary port. As a result of port 2 being a primary port used to receive data, the PAD 112 swaps the ports' MAC addresses. Once the MAC addresses are swapped, packets originally destined for port 1 are now received at port 2, since port 1 has failed. As a result of the MAC address swap, the PAD 112 calls an NDIS request function to update the handle and pointer of the change and hence reloads an offloaded connection. Office Action, pp. 4-5.

Respectfully, the Examiner's reasoning is incorrect on multiple counts. First, the Examiner's assertion that the PAD 112 calls the NDIS function and "reloads an offloaded connection" "as a result of the MAC address swap" is incorrect. The NDIS function is not called as a **result** of the MAC address swap; the NDIS function is the **means by which** the MAC address swap occurs at all. Col. 7, ll. 10-14 ("[PAD] 112 causes NID 103 to change the MAC addresses [of the ports] . . . [PAD] 112 does this by calling a Microsoft operating system function, called the NDIS request function.").

Second, the Examiner's assertion that the MAC addresses are swapped as a result of port 2 being designated a primary port also is incorrect. Burns does not appear to teach any such cause-and-effect relationship. Instead, Burns merely teaches that "[i]n addition to changing the primary team member, [PAD] 112 also swaps the MAC addresses of the two ports . . ." Col. 7, ll. 1-3.

Perhaps most significantly, Appellants note that the Examiner provides absolutely no reference to any teaching or suggestion in Burns that **anything** should happen as a result of receiving on port 2 a packet that is associated with port 1. The Examiner does make a cursory reference to the fact that "packets originally destined to port 1 are received at port 2," Office Action, p. 5, but fails to show where Burns teaches or even suggests the limitation in question.

Because – as the Examiner admits – Congdon fails to teach the limitation in question, and because Burns fails to satisfy Congdon's deficiencies, the combination of Congdon and Burns fails to teach the limitation in question.

### 4. Examiner's response in the Final Office Action dated July 27, 2009

In the Final Office Action dated July 27, 2009, the Examiner maintained his rejections and provided a response to Appellants' arguments. In essence, the Examiner argued that Fig. 2 of Burns does teach, "wherein a program executing on the CPU reloads an offloaded connection established by the first network adapter onto the second network adapter as a result of one of a plurality of packets associated with the offloaded connection being received on the second network adapter," as claimed. More particularly, the Examiner first observes that Port 1, which is assigned MAC address A, receives data from the switch 105, while Port 2, which is assigned MAC address C, sends data to the switch 105. The Examiner then postulates that Port 1 may fail, causing the MAC addresses of the ports to be swapped so that Port 1 has MAC address C and Port 2 has MAC address A. Thus, Port 2 now receives packets.

However, the Examiner argues, the Network Interface Device 103 may still need to transmit packets to the switch 105. Normally, according to the Examiner, Port 2 would be used to transmit packets to the switch 105, but Port 2 is busy receiving packets because Port 1 is defunct and can no longer receive or transmit packets. Thus, the Examiner argues, the solution is to swap the ports' MAC addresses again so that Port 2 can again transmit packets to the switch 105. Essentially, while Port 1 is defunct, Port 2 performs both the transmitting and receiving functions. The Examiner asserts that this teaching satisfies the quoted limitation.

### 5. Appellants' Response

What the Examiner continues to misunderstand is that claim 1 requires the re-loading of an offloaded connection "**as a result of** one of a plurality of packets associated with the offloaded connection being received on the second network adapter" (emphasis added). In the context of the Examiner's scenario above (where Port 1 is defunct and Port 2 is receiving packets originally intended for Port 1), the swapping of MAC addresses that causes Port 2 to again become a transmitting port **must occur as a result of receiving a packet on Port 2 that**

**was originally intended for Port 1**. Burns does not teach this requirement and it is telling that the Examiner, after many rejections, has still been unable to point to where Burns teaches or even suggests this requirement. It is true that Port 2 can receive packets that were originally intended for Port 1 when Port 1 becomes defunct; however, Port 2's reception of these packets does not **cause** Port 2 to again become a transmitting port. Instead, according to the Examiner, the **cause** for Port 2 to again become a transmitting port is "[i]f the destination wants to transmit a packet back to port 2 . . ." Final Office Action, p. 3, l. 10. Appellants respectfully ask the Board to focus in particular on the aspect of **causation** when comparing claim 1 to the art of record. This aspect substantially differentiates both the functionality and the purpose of Burns from those of the system in claim 1.

### C. The Examiner erred in rejecting claim 5 at least because the combination of Congdon, Burns and Mahalingham fails to disclose all claim limitations

The Examiner rejected claim 5 as allegedly obvious in view of Congdon, Burns and Mahalingham. Appellants traverse this rejection. As explained above, the Examiner erred in rejecting claim 1 using the combination of Congdon and Burns. Mahalingham fails to satisfy the deficiencies of this combination. Thus, the Examiner erred in rejecting claim 1 and all claims dependent on claim 1 (including claim 5) using the combination of Congdon, Burns and Mahalingham.

### D. Conclusion

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/Nick P. Patel/

Nick P. Patel
PTO Reg. No. 57,365
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
AGENT FOR APPELLANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
3404 E. Harmony Road
Fort Collins, CO 80528-9599

## VIII. CLAIMS APPENDIX

1. A computer system comprising:

a central processing unit (CPU); and

first and second network adapters teamed together and configured to receive offloaded connections;

wherein a program executing on the CPU reloads an offloaded connection established by the first network adapter onto the second network adapter as a result of one of a plurality of packets associated with the offloaded connection being received on the second network adapter.

2. The system of claim 1 wherein the first and second network adapters are capable of fully offloading all protocol processing.

3. The system of claim 1 wherein the first and second network adapters transmit and receive packets of data using a single media access control (MAC) and internet protocol (IP) address.

4. The system of claim 1 wherein the program reloads an offloaded connection by transferring the context of the connection from the first network adapter to the second network adapter.

5. The system of claim 1 wherein the program monitors every packet received by the first and second network adapters and inactivates connections associated with packets that have not been received for a defined time period.

6. The system of claim 1 wherein the first and second network adapters send a notification to the program if a connection is prematurely terminated.

7. The system of claim 1 wherein the first and second network adapters comprise network interface cards (NICs).

8. A method comprising:

examining a packet received from an external device;

determining whether a connection associated with the packet is currently offloaded; and

reloading the connection in response to the packet associated with the connection being offloaded and received by a network interface not currently processing the offloaded connection.

9. The method of claim 8 further comprising determining an identifier for the network interface that receives the packet and writing the determined identifier to a memory.

10. The method of claim 8 wherein the reloading further comprises copying the context of the connection to the network interface that received the packet.

11. The method of claim 8 wherein the network interface that received the packet and the network interface currently offloading the connection are teamed together.

12. A computer readable media storing instructions executable by a computer system, and when executed the instructions implement a method comprising:

examining a packet received from an external device;

determining whether a connection associated with the packet is currently offloaded; and

reloading the connection as a result of the packet associated with the connection being offloaded and received by a network interface not currently processing the offloaded connection.

13. The computer readable media of claim 12 further comprising determining an identifier for the network interface that receives the packet and writing the determined identifier to a memory unit.

14. The computer readable media of claim 12 wherein the reloading further comprises copying the context of the connection to the network interface that received the packet.

15. The computer readable media of claim 12 wherein the network interface that received the packet and the network interface currently offloading the connection are teamed together.

16. A computer system comprising:

means for reading and executing programs; and

first and second means for sending and receiving data connections over a network, the first and second means grouped together and capable of processing offloaded data connections;

wherein a program executed by the means for reading and executing programs reloads an offloaded connection established by the first means for sending and receiving data onto the second means for sending and receiving data in response to one of a plurality of packets associated with the offloaded connection being received on the second means for sending and receiving data.

17. The system of claim 16 wherein the first and second means for sending and receiving data connections are capable of fully offloading all protocol processing.

18. The system of claim 16 wherein the first and second means for sending and receiving data connections send and receive packets of data using a single media access control (MAC) and internet protocol (IP) address.

19. The system of claim 16 wherein the program reloads an offloaded connection by transferring the context of the connection from the first means for sending and receiving data connections to the second means for sending and receiving data connections.

20. The system of claim 16 wherein the program monitors all data received by the first and second means for sending and receiving data connections.

21. The system of claim 16 wherein the first and second means for sending and receiving data connections send a notification to the program if a connection is prematurely terminated.

## IX. EVIDENCE APPENDIX

None.

## X. RELATED PROCEEDINGS APPENDIX

None.